IoT WEB-SHARED VARIABLES – PUBLISH, COLLECT AND ANALYSIS IN THE CLOUD

Titus BĂLAN, Petruța BÎRLĂ, Cătălina MARCU, Ionela ONCERU

Transilvania University of Brasov, Romania, (titus.balan@unitbv.ro, petruta-p.birla@student.unitbv.ro, catalina.marcu@student.unitbv.ro, ionela.onceru@student.unitbv.ro)

DOI: 10.19062/1842-9238.2017.15.3.7

Abstract: *In a world where human activity tends to be the input data of IoT (Internet of Things) objects, AI (Artificial Intelligence) provides powerful tools for behavioral patterns with applications in wide domain areas. The approach described in this paper takes benefits of REST (Representational State Transfer) web-services to publish and get data to/from the Cloud, received from the IoT sensing elements. Based on the National Instruments web-shared variables server, the data is monitored and published. A Cloud-based IoT specific tool (Node-RED) is responsible for data retrieval using HTTP requests and extraction of the relevant values. Furthermore, these values are analyzed in order to create an AI service.*

Keywords: *IOT, AI, Cloud, ESP8266, Node-RED, REST services*

1. INTRODUCTION

Internet of Things has nowadays become a hot topic, not only as a research field, but as a growing industry, considering that more and more companies are developing businesses towards it: IBM, Google, Microsoft etc.

As technology develops in such an alert trend (in accordance to Moore's law), the number of devices with "intelligent" capabilities have increased: network connectivity, small scaled devices, even wearable technology, environment surveillance and interaction led to migration towards the Internet of Things. For example, sensor nodes can be connected to network and provide an environmental monitoring application: smart heating system, optimization of energy consumption etc. [1]

The communication between different "things" connected needs lightweight protocols, and some of the most well-known are MQTT and CoAP. Our demonstrator takes aid from REST services; as Web is implemented using RESTful principles, it would be common sense to utilize this method as means of communication. [2]

Furthermore, the Web Shared Variables server used in this demonstrator allocates resources dynamically and makes use of RESTful services, as seen above.

2. DATA TRANSMISSION AND COLLECTING ON SERVER

The scenario we have decided to implement in our demonstrator is a temperature monitoring solution, based on low-cost hardware elements that could benefit from the Cloud processing in the decision process for adapting the ambient temperature.

Similar, commercial products are already on the market (e.g. the Google Nest thermostat) but for exemplification purposes we have considered this to be a valid scenario, that could be enhanced with a more dense and accurate sensor network.

To transmit the acquired data from the temperature sensor we have chosen ESP8266.This is one of the cheapest Wi-Fi modules available on the market and can be implemented with the Arduino syntax. The ESP8266 processor from Espressif is an 80 MHz microcontroller with a full Wi-Fi front-end and TCP/IP stack. [3]

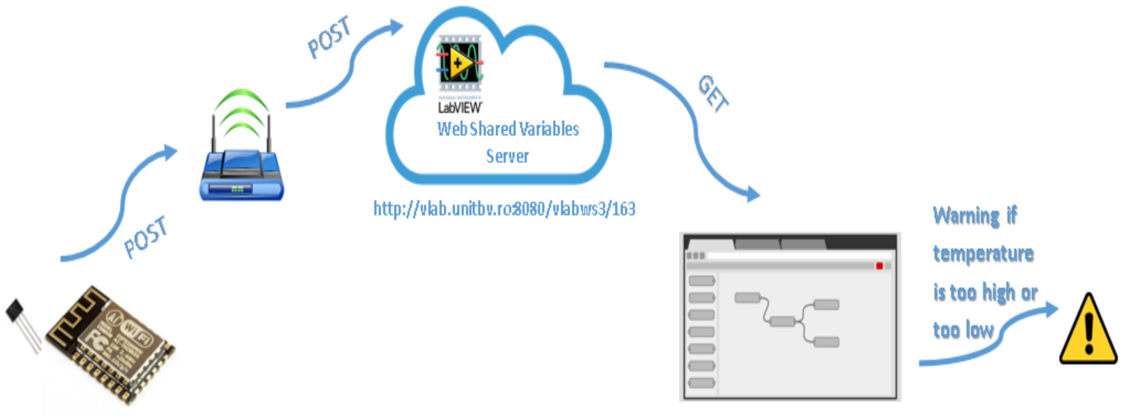

FIG. 1 Data transmission principle

Another used device that is connected to the development board is a LM35 temperature sensor. This is a device with a linear voltage to temperature in Celsius readout. It has a temperature range of -40 ° C up to 150 ° C and an output range from 0.1 V (-40 ° C) to 2.0 V (150 ° C). The development environment used to write the source code is the Arduino IDE program. This is an open source software that makes it easy to write a code and load it on the board.

The data acquisition made from the temperature sensor was sent to a shared variable server. The Shared Variable Server is an effective way to transmit data between multiple computers using simple encoding techniques and is installed with each version of LabView from version 8.0 upwards.

The Shared Variable Engine is where a variable is located on a local client. It is responsible for network communication, link management, and allows users to monitor the status of locally-implemented and real-time variables. Furthermore, since the server acts as a “man in the middle”, it can be considered as an IoT Gateway: it sits on the edge of the IoT ecosystem to provide data and connectivity throughout the communication. [4]

A communication process in LabView TCP / IP involves opening a connection, reading and / or writing information, and then closing the developed connection.

3. NODE-RED AS WORKING ENVIRONMENT

Probably one of the most specific software programs for IoT, Node-RED is a browser stream orchestrator that allows the connection / interconnection of IoT devices. Different services, APIs etc. can be applied to the resulting streams from these devices to create an entire system. [5]

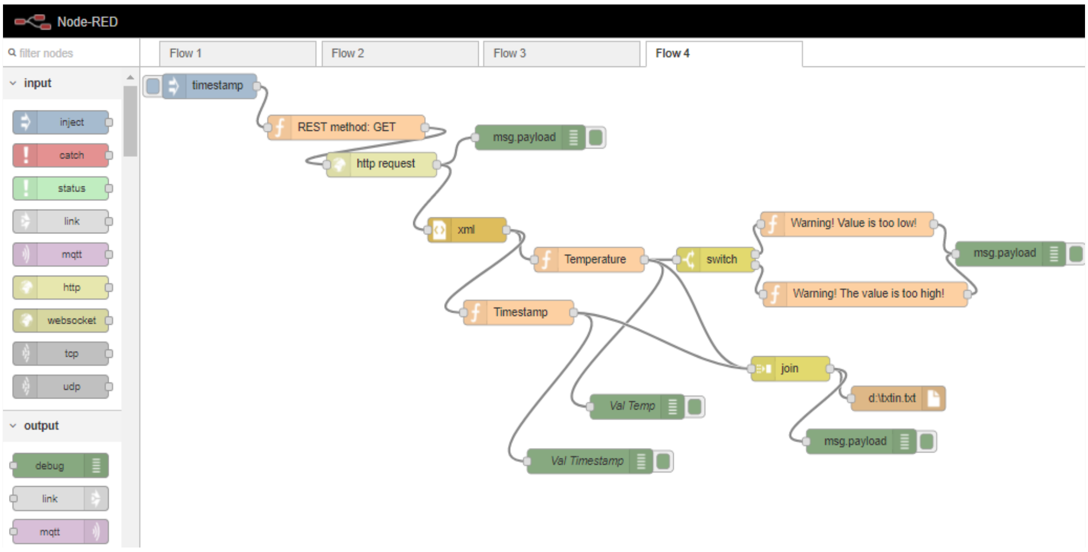

FIG. 2 Flow architecture in Node-RED

While REST stands for Representational State Transfer, which is an architectural style for networked hypermedia applications, it is primarily used to build Web services that are lightweight, maintainable, and scalable. A service based on REST is called a RESTful service. REST is not dependent on any protocol, but almost every RESTful service uses HTTP as its underlying protocol. Some might question our preference of using REST instead of IoT specific protocols like MQTT or CoAP, but as our application does not require at this point security barriers and because of the similarity of REST and MQTT werbs (POST – PUBLISH, GET - SUBSCRIBE) we have chosen REST web service. [6]

What is more, Node-RED can be successfully used to import "node" models to develop Artificial Intelligence services such as: machine learning, deep learning etc. for IoT.

4. DATA ANALYSIS

Retrieving data in Node-RED from the server is done using an HTTP request using a GET method. This method interrogates data from the server address: http://vlab.unitbv.ro:8080/vlabws3/163. The data is then displayed using a Debug Node:

```
<Response>
  <Terminal>
    <Name>Temperature</Name>
    <Value>24.4</Value>
  </Terminal>
  <Terminal>
    <Name>Device Time</Name>
    <Value>11/19/2017 7:38 PM</Value>
  </Terminal>
</Response>
```


The Inject node (timestamp) allows to inject messages into a flow, either by clicking the button on the node, or setting a time interval between injects (the repeat interval is at every 15 minutes).

As URL is used to identify resources over the Internet and HTTP as a service interface, an IoT "object" can thus connect to a certain service, located at the address pointed out by URL and send HTTP calls. The format used in this experiment is XML, by which the payload data is exchanged using GET, POST, PUT etc. verbs.

Because the data retrieved from the HTTP request is in string format, it must then be converted for easier processing in XML format using an XML node. To extract the temperature value from the data block, is used the following function: "msg.payload = msg.payload.Response.Terminal[0].Value[0]; return msg;". Then it is displayed, by using a Debug Node.

For temperature processing, is used a switch function, that route messages based on their property values: if temperature is less than 22 degrees, then will be displayed a warning that contains the following message: "Small Value"; if temperature is greater than 23 degrees, will be displayed the following message: "Exceeded Value".

Also, the value of timestamp is extracted from the data block, using the following function: "msg.payload = msg.payload.Response.Terminal[1].Value[0]; return msg;".

The value of timestamp and temperature are joined together in a file, that could potentially be further needed in a machine learning service as described in conclusions paragraph.

5. CLOUD - EDGE PARADIGM IN DATA ANALYSIS

Recent years, Cloud Computing has been on a rising trend for, not only storage, but for data processing. IoT had nevertheless took advantage of Cloud by providing MQTT brokers, for example. Node-RED described above can be classified as a Cloud solution for IoT stream orchestrator. [7]

However, if the incident data has critical, real-time characteristics, then the Cloud architecture impacts on both accuracy of data and speed of delivery, such it imposes an Edge approach.

Node-RED, as a lightweight platform can also be used as an Edge processor: if the IoT object is an SBC (Single Board Computer), it can process the time critical data itself.

6. CONCLUSIONS AND FURTHER DEVELOPMENT

In the above presented scenario we were able to demonstrate the easy methods to integrate sensing data into REST-based IoT processing flows, that can further become complex services. The advantage of the presented system is the cost (by using a low-cost processor) and the easy integration with other applications based on web services. We have proved the efficiency of Node-RED that could also be an excellent inter-protocol mediator for different IoT protocols (e.g. MQTT interconnection with REST/HTTP), acting as a broker: can implement himself some of the decisions, but can also pass values to standard Cloud platforms from Amazon, Microsoft or IBM.

Our implementation has potential to develop machine learning services from the incoming data from the server. One potential scenario could involve AAL (Active Assisted Living) services: sensors that monitors day-to-day activity of elderly people and “learn” habits to prevent any anomaly that could be a sign of illness or disease. [8]

The flow below can be considered as a starting point as a future development.

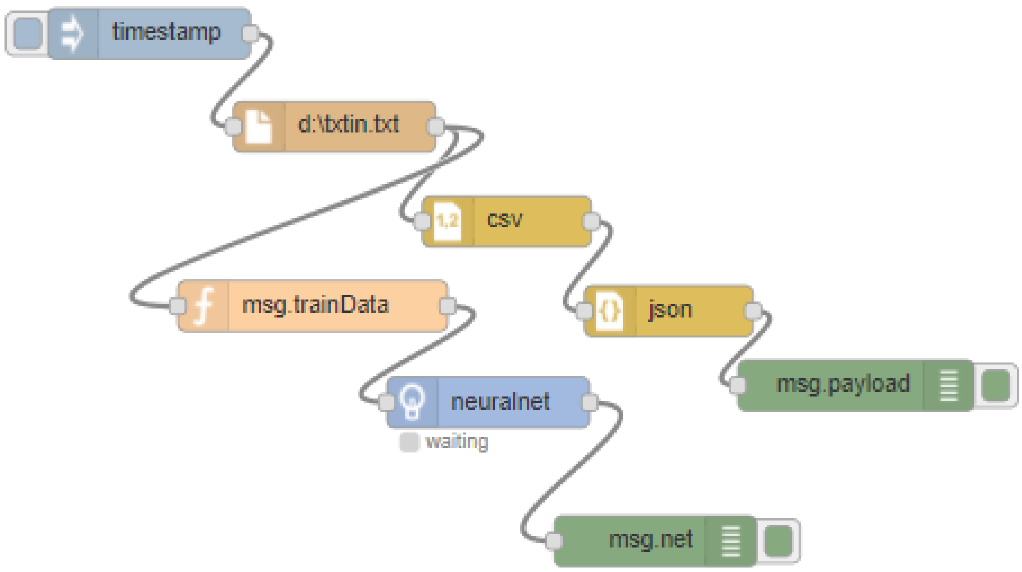

FIG. 3 Flow architecture for further development

Cloud industry provides more and more tools and services in this direction, a future development could take into consideration IBM Watson services or Azure IoT Hub logic connectivity programming.

REFERENCES

[1] John Soldatos, Nikos Kefalakis, Manfred Hauswirth, Martin Serrano, “OpenIoT: Open Source Internet-of-Things in the Cloud”, 2015, https://www.researchgate.net/publication/273447891_OpenIoT_Open_Source_Internet-of-Things_in_the_Cloud;

[2] L. Xiao-Hong, "Research and Development of Web of Things System Based on Rest Architecture," 2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications, Hunan, 2014, pp. 744-747;

[3] “ESP8266 FOTA Introduction”, http://www.espressif.com/sites/default/files/99c-esp8266_ota_upgrade_en_ v1.6.pdf, [accessed in Nov. 2017];

[4] D.N.Robu, P.V.Fernoagă, G.A.Stelea, F.Sandu – ”Tele-measurement with Virtual Instrumentation using Web-Services” - Peoceedings of the IEEE 23rd International Symposium for Design and Technology in Electronic Packaging, "SIITME", Constanta-RO, Oct.26-28, 2017;
[5] https://nodered.org/, [accessed in Nov. 2017];
[6] D. Guinard, I. Ion, and S. Mayer “In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers’ Perspective” - https://www.vs.inf.ethz.ch/publ/papers/dguinard-rest-vs-ws.pdf [accessed in Oct. 2017];
[7] D.N.Robu and C.Costache (editors) - Inteligent Telecommunication Networks - "Transilvania" publishing house, Brasov-RO, 2016, ISBN 978-606-19-0820-2;
[8] Sorin-Aurel Moraru, Liviu Perniu, Dominic Mircea Kristaly, Delia A. Ungureanu, Florin Sandu, Adrian A. Misoi, “Integrating wireless sensors into cloud systems for ambient assisted living,”, http://ieeexplore.ieee.org/document/7984266/ [accessed in Nov. 2017].